Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 11 of 11 returned.

1. Document ID: US 6654017 B1

L14: Entry 1 of 11 File: USPT Nov 25, 2003

DOCUMENT-IDENTIFIER: US 6654017 B1
TITLE: Smoothing of line segments on a computer display

Brief Summary Text (9):
Typical supersampling techniques attempt to smooth the staircase effect by using the surrounding pixels. FIG. 2 is a diagram illustrating a supersampling technique applied to the aliased line of FIG. 1. The group of shaded pixels 200 is now much wider than in FIG. 1, and there is less of a staircase effect simply because they are wider. However, this technique also utilizes the ability to vary intensity of each pixel to smooth the edges of the line. For example, without the ability to vary intensities, pixel 202 would have to be completely shaded. However, by selectively turning on or off certain subpixels within pixel 202, it give the viewer the impression that only a region of pixel 202 is filled, and thus makes the line appear much smoother.

Drawing Description Text (3):
FIG. 2 is a diagram illustrating a supersampling technique applied to the aliased line of FIG. 1.

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw De

2. Document ID: US 6636231 B1

L14: Entry 2 of 11 File: USPT Oct 21, 2003

DOCUMENT-IDENTIFIER: US 6636231 B1
TITLE: Apparatus for approximation of caps of smooth line segments

Brief Summary Text (9):
Typical supersampling techniques attempt to smooth the staircase effect by using the surrounding pixels. FIG. 2 is a diagram illustrating a supersampling technique applied to the aliased line of FIG. 1. The group of shaded pixels 50 is now much wider than in FIG. 1, and there is less of a staircase effect simply because they are wider. However, this technique also utilizes the ability to vary intensity of each pixel to smooth the edges of the line. For example, without the ability to vary intensities, pixel 52 would have to be completely shaded. However, by varying the intensity of pixel 52, it gives the viewer the impression that only a region of

pixel 52 is filled, and thus makes the line appear much smoother.

Drawing Description Text (3):
FIG. 2 is a diagram illustrating a supersampling technique applied to the aliased line of FIG. 1.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw. De

3. Document ID: US 6636230 B1

L14: Entry 3 of 11 File: USPT Oct 21, 2003

DOCUMENT-IDENTIFIER: US 6636230 B1
TITLE: Method for approximation of caps of smooth line segments

Brief Summary Text (9):
Typical supersampling techniques attempt to smooth the staircase effect by using the surrounding pixels. FIG. 2 is a diagram illustrating a supersampling technique applied to the aliased line of FIG. 1. The group of shaded pixels 50 is now much wider than in FIG. 1, and there is less of a staircase effect simply because they are wider. However, this technique also utilizes the ability to vary intensity of each pixel to smooth the edges of the line. For example, without the ability to vary intensities, pixel 52 would have to be completely shaded. However, by varying the intensity of pixel 52, it gives the viewer the impression that only a region of pixel 52 is filled, and thus makes the line appear much smoother.

Drawing Description Text (3):
FIG. 2 is a diagram illustrating a supersampling technique applied to the aliased line of FIG. 1.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw. De

4. Document ID: US 6452595 B1

L14: Entry 4 of 11 File: USPT Sep 17, 2002

DOCUMENT-IDENTIFIER: US 6452595 B1
TITLE: Integrated graphics processing unit with antialiasing

Detailed Description Text (211):
In accordance with the operation of known supersampling techniques, the color value and Z-value for each reference "sample" point within a pixel are first sent to a processor which computes the color-value for each sample and compares the Z-value of each covered sample against any previously stored value. The color value for each newly covered sample is then updated accordingly.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw. De

5. Document ID: US 6204859 B1

L14: Entry 5 of 11 File: USPT Mar 20, 2001

DOCUMENT-IDENTIFIER: US 6204859 B1
TITLE: Method and apparatus for compositing colors of images with memory constraints for storing pixel data

Detailed Description Text (15):
Accordingly, recent modern implementations do not sample at every subpixel 206. Rather, those subpixels 206 which are sampled are sparsely distributed in the subpixel array. In general, the antialiasing results were almost as effective for such sparse supersampling as for the full scene supersampling technique.

Detailed Description Text (31):
In one embodiment, described further in connection with FIG. 6A, the linking is accomplished by storing an index value for each subpixel sample S1-S4. Accordingly, this embodiment is coined indexed sparse supersampling. In another embodiment, described in connection with FIG. 6C, the linking is accomplished by storing a coverage mask, or bit pattern, for each stored fragment value. This embodiment is hereafter referred to as an improved A-buffer technique. Collectively, the embodiments are referred to as improved supersampling techniques.

Detailed Description Text (36):
Rather than storing four fragment triples in the pixel memory 314, one for each of the four subpixel samples S1-S4, which would be done using typical supersampling techniques, the exemplary embodiment in FIG. 3 stores only two fragment triples 310, 312. Accordingly, the invention avoids storing redundant data for the pixel 300 because only one instance of the fragment triple 310 is stored for the three subpixel samples S2-S4. By so doing, the storage requirements for fragment triples are considerably reduced.

Detailed Description Text (37):
For example, if each fragment triple 310, 312 requires nine bytes of storage, then the improved supersampling techniques use approximately eighteen bytes of memory per pixel fewer than typical supersampling methods. The improved supersampling techniques do use additional memory for storing the pointers 320-326, but this amount is small when compared to the memory saved by storing only two fragment triples 310, 312 for the four subpixel samples S1-S4.

Detailed Description Text (38):
The memory savings increase when the pixel 300 is subdivided into one of the larger subpixel arrays 202, 204. With the 8.times.8 subpixel array 202 and the sampling pattern 220 (N equals 8), the improved supersampling techniques use fifty-four fewer bytes per pixel than typical supersampling. This is because only two of eight sampled fragment triples are stored in the pixel memory 314. For the 16.times.16 subpixel array 204 and the sampling pattern 230 (N equals 16), only two of sixteen sampled fragment triples are stored in the pixel memory 314, and so 112 bytes per pixel are saved. For a display screen 130 with 1920.times.1200 pixels, such savings amount to approximately 258 Mbytes.

Detailed Description Text (89):
FIGS. 6A and 6B illustrate a exemplary logical representation of the pixel memory 314 used by the indexed sparse supersampling technique. The pixel memory 314 includes indices 600 and fragment triples 310, 312. The pixel memory 314 provides storage for a predetermined number of fragment triples. Although shown to be

contiguous in the graphics memory 122, the indices 600 can be separate from each other or from the fragment triples 310, 312.

Detailed Description Text (94):
With one bit per index 602-608, the sampling pattern 210 (N=4) needs four bits of pixel memory 314 to implement the indices 600. The storage requirements for indices 600 of larger sampling patterns 220, 230 is also small. For example, the sampling pattern 230 (N=16) would need 16 bits per pixel 134 to implement one bit per index. Implementing four bits per index uses 64 bits per pixel, which still provides a sizable storage savings over typical sparse supersampling techniques that store sixteen fragment triples for the sixteen subpixels samples S1-S16.

Detailed Description Text (100):
For the sampling pattern 220, each coverage mask 620, 622, 624 requires eight bits, one for each of the eight subpixel samples S1-S8. As for the sampling pattern 230, which has sixteen subpixel samples S1-S16, there would be sixteen such bits in each coverage mask. Yet even with 16 bits per coverage mask, the storage savings are sizable over known sparse supersampling techniques that store sixteen fragment triples for the sixteen subpixels samples S1-S16.

Detailed Description Text (109):
The exemplary embodiments shown in FIGS. 6A-6D can achieve satisfactory antialiasing results by storing two fragment triples for four subpixel samples. Eight subpixel samples with two stored fragment triples usually looks better than four subpixel samples with two fragment triples, but can look worse when one of the additional four subpixel samples requires replacing one of the stored triples with a third fragment triple, and that third fragment triple appears in the pixel memory last. Thus, allocating storage for a third fragment triple can make a marked improvement for eight subpixel samples over storing two fragment triples. Clearly, the antialiasing results can be made to approach the results of typical sparse supersampling as more fragment triples are stored, but each additional triple erodes the memory savings provided by the improved supersampling techniques.

Detailed Description Text (113):
When the new fragment is visible at one of the covered subpixel samples, then the graphics accelerator 108 invalidates the link between each covered sample and a stored fragment, if the new fragment obscures the stored fragment for that covered subpixel sample. For the indexed sparse supersampling technique, the graphics accelerator 108 maintains control bits for keeping track of the validity of each index and invalidates each index linking a covered subpixel sample to an obscured fragment. The control bits may direct the graphics accelerator 108 to use the default background color if no fragments cover a subpixel sample. For the improved A-buffer technique, the bits in the coverage mask associated with each covered subpixel sample are unchanged when the new fragment is transparent and are set to "0" when the new fragment is opaque.

Detailed Description Text (114):
Then, in step 708, the number of links pointing to each fragment triple is counted. For the indexed sparse supersampling technique, step 708 counts the number of indices linked to that stored fragment triple. For the improved A-buffer technique, step 708 counts the number of bits in each coverage mask that have a "1" bit value.

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw De

6. Document ID: US 6128000 A

L14: Entry 6 of 11 File: USPT Oct 3, 2000

DOCUMENT-IDENTIFIER: US 6128000 A
TITLE: Full-scene antialiasing using improved supersampling techniques

Brief Summary Text (11):
In accordance with the present invention, an objective is to minimize aliasing artifacts in graphic images using improved supersampling techniques. Another objective is to effectively minimize these artifacts while reducing the memory capacity and bandwidth requirements associated with prior art antialiasing techniques.

Detailed Description Text (15):
Accordingly, recent modern implementations do not sample at every subpixel 206. Rather, those subpixels 206 which are sampled are sparsely distributed in the subpixel array. In general, the antialiasing results were almost as effective for such sparse supersampling as for the full scene supersampling technique.

Detailed Description Text (31):
In one embodiment, described further in connection with FIG. 6A, the linking is accomplished by storing an index value for each subpixel sample S1-S4. Accordingly, this embodiment is coined indexed sparse supersampling. In another embodiment, described in connection with FIG. 6C, the linking is accomplished by storing a coverage mask, or bit pattern, for each stored fragment value. This embodiment is hereafter referred to as an improved A-buffer technique. Collectively, the embodiments are referred to as improved supersampling techniques.

Detailed Description Text (37):
Rather than storing four fragment triples in the pixel memory 314, one for each of the four subpixel samples S1-S4, which would be done using typical supersampling techniques, the exemplary embodiment in FIG. 3 stores only two fragment triples 310, 312. Accordingly, the invention avoids storing redundant data for the pixel 300 because only one instance of the fragment triple 310 is stored for the three subpixel samples S2-S4. By so doing, the storage requirements for fragment triples are considerably reduced.

Detailed Description Text (38):
For example, if each fragment triple 310, 312 requires nine bytes of storage, then the improved supersampling techniques use approximately eighteen bytes of memory per pixel fewer than typical supersampling methods. The improved supersampling techniques do use additional memory for storing the pointers 320-326, but this amount is small when compared to the memory saved by storing only two fragment triples 310, 312 for the four subpixel samples S1-S4.

Detailed Description Text (39):
The memory savings increase when the pixel 300 is subdivided into one of the larger subpixel arrays 202, 204. With the 8.times.8 subpixel array 202 and the sampling pattern 220 (N equals 8), the improved supersampling techniques use fifty-four fewer bytes per pixel than typical supersampling. This is because only two of eight sampled fragment triples are stored in the pixel memory 314. For the 16.times.16 subpixel array 204 and the sampling pattern 230 (N equals 16), only two of sixteen sampled fragment triples are stored in the pixel memory 314, and so 112 bytes per pixel are saved. For a display screen 130 with 1920.times.1200 pixels, such savings amount to approximately 258 Mbytes.

Detailed Description Text (92):
FIGS. 6A and 6B illustrate a exemplary logical representation of the pixel memory

314 used by the indexed sparse supersampling technique. The pixel memory 314 includes indices 600 and fragment triples 310, 312. The pixel memory 314 provides storage for a predetermined number of fragment triples. Although shown to be contiguous in the graphics memory 122, the indices 600 can be separate from each other or from the fragment triples 310, 312.

Detailed Description Text (97):
With one bit per index 602-608, the sampling pattern 210 (N=4) needs four bits of pixel memory 314 to implement the indices 600. The storage requirements for indices 600 of larger sampling patterns 220, 230 is also small. For example, the sampling pattern 230 (N=16) would need 16 bits per pixel 134 to implement one bit per index. Implementing four bits per index uses 64 bits per pixel, which still provides a sizable storage savings over typical sparse supersampling techniques that store sixteen fragment triples for the sixteen subpixels samples S1-S16.

Detailed Description Text (103):
For the sampling pattern 220, each coverage mask 620, 622, 624 requires eight bits, one for each of the eight subpixel samples S1-S8. As for the sampling pattern 230, which has sixteen subpixel samples S1-S16, there would be sixteen such bits in each coverage mask. Yet even with 16 bits per coverage mask, the storage savings are sizable over known sparse supersampling techniques that store sixteen fragment triples for the sixteen subpixels samples S1-S16.

Detailed Description Text (112):
The exemplary embodiments shown in FIGS. 6A-6D can achieve satisfactory antialiasing results by storing two fragment triples for four subpixel samples. Eight subpixel samples with two stored fragment triples usually looks better than four subpixel samples with two fragment triples, but can look worse when one of the additional four subpixel samples requires replacing one of the stored triples with a third fragment triple, and that third fragment triple appears in the pixel memory last. Thus, allocating storage for a third fragment triple can make a marked improvement for eight subpixel samples over storing two fragment triples. Clearly, the antialiasing results can be made to approach the results of typical sparse supersampling as more fragment triples are stored, but each additional triple erodes the memory savings provided by the improved supersampling techniques.

Detailed Description Text (116):
When the new fragment is visible at one of the covered subpixel samples, then the graphics accelerator 108 invalidates the link between each covered sample and a stored fragment, if the new fragment obscures the stored fragment for that covered subpixel sample. For the indexed sparse supersampling technique, the graphics accelerator 108 maintains control bits for keeping track of the validity of each index and invalidates each index linking a covered subpixel sample to an obscured fragment. The control bits may direct the graphics accelerator 108 to use the default background color if no fragments cover a subpixel sample. For the improved A-buffer technique, the bits in the coverage mask associated with each covered subpixel sample are unchanged when the new fragment is transparent

Detailed Description Text (118):
Then, in step 708, the number of links pointing to each fragment triple is counted. For the indexed sparse supersampling technique, step 708 counts the number of indices linked to that stored fragment triple. For the improved A-buffer technique, step 708 counts the number of bits in each coverage mask that have a "1" bit value.

7. Document ID: US 6072500 A

L14: Entry 7 of 11 File: USPT Jun 6, 2000

DOCUMENT-IDENTIFIER: US 6072500 A
TITLE: Antialiased imaging with improved pixel supersampling

Brief Summary Text (6):
In accordance with the operation of prior art supersampling techniques, the color value and Z-value for each reference point within a pixel are first sent to a processor which computes the color-value for each super-sample and compares the Z-value of each covered supersample against any previously stored value. The color value for each newly covered supersample is then updated accordingly. Supersampling using this prior art approach requires a substantial amount of processor performance, as well as storage space. The number of memory accesses required using this prior art technique is particularly notable, especially when it is considered that it is not only necessary to compute and store color values and Z-values for each of the supersamples (ie., N.times.1.3 million pixels of a 1024.times.1280 array, where N is the number of data samples taken per pixel), but that it is also necessary to update color-values and Z-values for each of the newly covered supersamples. Although the computation of color-values and Z-values for each supersample can be parallelized to speed up the process, at the cost of adding expensive processors, updating both of these values for every super-sample nevertheless becomes a significant performance limiting consideration.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

8. Document ID: US 5877771 A

L14: Entry 8 of 11 File: USPT Mar 2, 1999

DOCUMENT-IDENTIFIER: US 5877771 A
TITLE: Method and apparatus for supersampling based on the local rate of change in texture

Brief Summary Text (13):
Supersampling techniques have been developed to improve the texture detail which is mapped to a pixel footprint area. In so-called regular supersampling, an image is sampled n times the display resolution to improve the texture display quality (see FIG. 1B, n=9 adapted from Wolberg at page 169). Irregular supersampling techniques take samples at non-uniform sampling locations to further reduce aliasing artifacts which can arise when texture samples are regularly-spaced. See, e.g, the discussion of irregular sampling techniques such as stochastic sampling including Poisson sampling, jittered sampling, and point-diffusion sampling, in Wolberg, pp. 173-177. Regular and irregular supersampling techniques, however, have been inefficient and unworkable for real-time display of special effects images. Computational costs and delays quickly become prohibitive, especially when images are distorted, rotated, warped, or displayed in perspective.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

☐ 9. Document ID: US 5742277 A

L14: Entry 9 of 11 File: USPT Apr 21, 1998

DOCUMENT-IDENTIFIER: US 5742277 A
TITLE: Antialiasing of silhouette edges

Brief Summary Text (17):
Thus, less expensive computer graphics systems cannot employ supersampling techniques to alleviate aliasing. Instead, such less expensive computer graphics systems typically employ non-supersampling techniques such as shown in FIG. 1. Accordingly, the visual quality of lower end, less expensive computer graphics systems is relatively low.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

☐ 10. Document ID: US 5684939 A

L14: Entry 10 of 11 File: USPT Nov 4, 1997

DOCUMENT-IDENTIFIER: US 5684939 A
TITLE: Antialiased imaging with improved pixel supersampling

Brief Summary Text (6):
In accordance with the operation, of prior art supersampling techniques, the color value and Z-value for each reference point within a pixel are first sent to a processor which computes the color-value for each supersample and compares the Z-value of each covered supersample against any previously stored value. The color value for each newly covered supersample is then updated accordingly. Supersampling using this prior art approach requires a substantial amount of processor performance, as well as storage space. The number of memory accesses required using this prior art technique is particularly notable, especially When it is considered that it is not only necessary to compute and store color values and Z-values for each of the supersamples (i.e., N.times.1.3 million pixels of a 1024.times.1280 array, where N is the number of data samples taken per pixel), but that it is also necessary to update color-values and Z-values for each of the newly covered supersamples. Although the computation of color-values and Z-values for each supersample can be parallelized to speed up the process, at the cost of adding expensive processors, updating both of these values for every supersample nevertheless becomes a significant performance limiting consideration.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw De

☐ 11. Document ID: US 5019903 A

L14: Entry 11 of 11 File: USPT May 28, 1991

DOCUMENT-IDENTIFIER: US 5019903 A
TITLE: Spatial interpolation between lines of a supersampled digital video signal in accordance with a gradient vector selected for maximum matching of blocks of samples which are offset in opposite directions

Drawing Description Text (6):
FIGS. 7 to 9 are diagrams used to explain a supersampling technique used in the preferred method and apparatus embodying the invention;

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw. De

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Terms	Documents
supersampling technique	11

Display Format: KWIC Change Format

Previous Page Next Page Go to Doc#